REMARKS

1. Objections of drawings under 37 CFR 1.83(a):

Examiner stated that the features of the invention specified in the claims: "a portion of the memory", "updating is performed before the BIST performed", "completely through with the entirety of the memory" and "a first and a second defective portion of the memory" must be shown in the drawings.

Response:

As to the feature "a portion of the memory":

As stated in the paragraph [0004], "...A packet buffer is usually segmented into sub-blocks, where these sub-blocks are designated as "pages"...," since this is very well known in the art, the details of the packet buffer are not shown in the drawings. The paragraph [0004] further stats, "... Conceptually, a linked list contains entries, wherein **each entry is associated with one page of the packet buffer**". Additionally, the paragraph [0005] states "As shown in Fig.1, the linked list with 8 entries includes a first pointer field corresponding to the pointers to the current page of an associated packet buffer (not shown), and a second pointer field corresponding to the entry associated with the next page. **Take Entry 4 as an example; it is shown in Fig.1 that the first pointer of Entry 4 corresponds to a current page (Page 4) of the packet buffer, and the second pointer of Entry 4 corresponds to the entry associated with the next page, which is Entry 5 in this case.** This same manner of association

can be seen for each entry of the linked list". Accordingly, applicants believe that the feature "a portion of the memory" specified in claim 13 can be readily observed from Fig. 1 and supported by above-identified paragraphs.

As to the features "updating is performed before the BIST performed" and "completely through with the entirety of the memory":

Applicants believe that the Examiner misunderstood these two features. These two features are specified in claim 16, **but they should be combined together as a timing feature of the update operation of the linked list** rather than separated into two features. The claim 16 specifies a feature: "**the step (c) of updating is performed before the BIST performed in step (b) is completely through with the entirety of the memory**". This feature is supported by the flowchart shown in Fig. 3 and paragraphs [0021]-[0027]. Specifically, the switch 10 performs a BIST on the header table 20 (Step 110) to determine which sections of the header table 20 are defective (see paragraph [0021] lines 1-4), and dynamically updates the linked list (Step 120).

As described in paragraph [0022], "... **each time a defective section is found in the header table** 20, the **switch 10 will pause the BIST, update the linked list dynamically so as not to use the defective section in the list,** and **then continue the BIST**...". This paragraph clearly shows the "dynamic" nature of the update operation of the linked list disclosed by the present invention. Therefore, Step 120 of Fig. 3 is designated as "Dynamic update" to point out this feature.

According to the above descriptions, it can be readily derived that the switch 10 **alternately** performs Step 110 and Step 120 until all sections of the header table 20 are completely tested. In other words, the operation of updating the linked list is performed **before** the entirety of the header table 20 is completely tested by the BIST operation of Step 110.

Then, the switch 10 performs a BIST on the packet buffer 30 (Step 130) to determine which pages of the packet buffer 30 are defective, and dynamically updates the linked list (Step 140). As the descriptions in paragraph [0024] lines 2-7 state, "each time a defective page is found in the packet buffer 30, the switch 10 will **pause the BIST**, update the linked list dynamically so as not to use the section corresponding to the defective page of the packet buffer 30, and **then continue the BIST**," this paragraph again shows the "dynamic" nature of the update operation of the linked list disclosed by the present invention. Additionally, Step 140 of Fig. 3 is designated as "Dynamic list update" to highlight this feature.

Similar to Steps 110 and 120, the switch 10 also performs Step 130 and Step 140 **alternately** until all pages of the packet buffer 30 are completely tested. In other words, the operation of updating the linked list is performed **before** the entirety of the packet buffer 30 is completely tested by the BIST operation of Step 130.

In accordance with the above explanations, it can be appreciated that the feature "**the step (c) of updating is performed before the BIST performed in step (b) is completely through with the entirety of the memory**" specified in claim 16 can be readily observed from the flowchart illustrated in Fig. 3.

As to the feature "a first and a second defective portion of the memory":

As stated in paragraph [0019] lines 7-11, a reference, "a programmable BIST core for embedded DRAM" by Huang et al. concerning the details of the built-in self test (BIST), is incorporated by reference with the present application. This paragraph clearly states that the way to detect defective portions of the memory by using BIST is very well known in the art. Furthermore, a defective portion of the memory is just a portion of the memory, such as a page, and it is not essential for a proper understanding of the invention. It can be readily appreciated that the packet buffer 30 may have two or more defective pages. Therefore, the features "a first defective portion of the memory" and "a second defective portion of the memory" specified in claims are intended to clearly highlight the "dynamic" nature of the operation of updating the linked list when the memory has two defective portions. Applicants therefore believe that the feature "a first defective portion of the memory" specified in claims 13 and 22 and the feature "a second defective portion of the memory" specified in claims 17 and 22 are not necessary to be shown in the drawings.

2. Rejections of claims under 35 U.S.C. 112:

Claims 13-18 and 22 are rejected under 35 U.S.C. 112, first paragraph, as failing comply with the written description requirement.

Response:

Claim 13

Examiner stated that the limitations "wherein each entry of the linked list corresponds to a portion of the memory", "a portion of the memory", and "a first defective portion of the memory" in claim 13 were not described in the specification at the time the application was filed.

The paragraph [0004] lines 4-5 state, "A packet buffer is usually segmented into sub-blocks, where these sub-blocks are designated as "pages". The paragraph [0004] lines 9-10 state, "Conceptually, a linked list contains entries, wherein **each entry is associated with one page of the packet buffer**". Additionally, the descriptions in paragraph [0005] lines 2-12 state "As shown in Fig.1, the linked list with 8 entries includes a first pointer field corresponding to the pointers to the current page of an associated packet buffer (not shown), and a second pointer field corresponding to the entry associated with the next page. **Take Entry 4 as an example; it is shown in Fig.1 that the first pointer of Entry 4 corresponds to a current page (Page 4) of the packet buffer, and the second pointer of Entry 4 corresponds to the entry associated with the next page, which is Entry 5 in this case.** This same manner of association can be seen for each entry of the linked list". Accordingly, applicants believe that the feature "wherein each entry of the linked list corresponds to a portion of the memory" and "a portion of the memory" specified in claim 13 can be readily observed from Fig. 1 and supported by above-identified paragraphs.

In addition, the feature "a first defective portion of the memory" can be readily observed from the paragraph [0024] lines 2-7. In this paragraph, "each time a defective page is found in the packet buffer 30..." means

"each time **a defective page of the packet buffer 30** is found". It is obvious that the feature "a first defective portion of the memory" is fully supported by the above-identified paragraph.

Claim 16

Examiner stated that the limitations "updating is performed before the BIST performed" and "completely through with the entirety of the memory" in claim 16 were not described in the specification at the time the application was filed.

As explained above, these two features **should be combined together as a timing feature of the update operation of the linked list** rather than separated into two features. The whole limitation is "**the step (c) of updating is performed before the BIST performed in step (b) is completely through with the entirety of the memory**". This feature is supported by the flowchart shown in Fig. 3 and the paragraphs [0021]-[0027]. According to the paragraphs [0021] and [0022], the switch 10 performs a BIST on the header table 20 (Step 110) to determine which sections of the header table 20 are defective (see paragraph [0021] lines 1-4), and dynamically updates the linked list (Step 120).

As described in paragraph [0022] lines 3-7, "**each time a defective section is found in the header table** 20, the **switch 10 will pause the BIST, update the linked list dynamically so as not to use the defective section in the list,** and **then continue the BIST**," this paragraph clearly shows the "dynamic" nature of the update operation of the linked list disclosed by the present invention. Step 120 of Fig. 3 is designated as

"Dynamic update" to point out this feature.

According to the above descriptions, it can be readily derived that the switch 10 **alternately** performs Step 110 and Step 120 until all sections of the header table 20 are completely tested. In other words, the operation of updating the linked list is performed **before** the entirety of the header table 20 is completely tested by the BIST operation of Step 110.

Then, the switch 10 performs a BIST on the packet buffer 30 (Step 130) to determine which pages of the packet buffer 30 are defective, and dynamically updates the linked list (Step 140). As the descriptions in paragraph [0024] lines 2-7 state, "each time a defective page is found in the packet buffer 30, the switch 10 will **pause the BIST**, update the linked list dynamically so as not to use the section corresponding to the defective page of the packet buffer 30, and **then continue the BIST,**" this paragraph again shows the "dynamic" nature of the update operation of the linked list disclosed by the present invention. Step 140 of Fig. 3 is designated as "Dynamic list update" to highlight this feature.

Similarly, the switch 10 also performs Step 130 and Step 140 **alternately** until all pages of the packet buffer 30 are completely tested. In other words, the operation of updating the linked list is performed **before** the entirety of the packet buffer 30 is completely tested by the BIST operation of Step 130.

According to above explanations, it can be appreciated that the feature "**the step (c) of updating is performed before the BIST performed in step (b) is completely through with the entirety of the memory**" can be

readily observed from the flowchart illustrated in Fig. 3 and supported by the paragraphs [0021]-[0027].

Claim 22

Examiner stated that the limitation "a first and a second defective portion of the memory" in claim 22 was not described in the specification at the time the application was filed.

Applicants assert that this limitation can be readily observed from paragraph [0024] lines 2-7. As described in the identified paragraph, each time a defective page of the packet buffer 30 is found by the BIST in Step 130, the switch 10 pauses the BIST of Step 130, and performs Step 140 to dynamically update the linked list by excluding the use of the section of the header table 20 corresponding to the defective page of the packet buffer 30. Then, the switch 10 continues the BIST of Step 130 to determine if the packet buffer 30 has other defective pages. The switch 10 repeats the operations of Steps 130 and 140 until the linked list is updated to exclude the use of the sections of the header table 20 corresponding to the defective pages of the packet buffer 30 (see paragraph [0024] lines 7-9). In other words, if a next defective page of the packet buffer 30 is found by the BIST, the switch 10 performs similar operations as well as that when the previous defective page of the packet buffer 30 is found.

It can be readily appreciated that the packet buffer 30 may have two or more defective pages. The limitations "a first defective portion of the memory" and "a second defective portion of the memory" in claim 22 simply highlight the "dynamic" nature of the operation of updating the

linked list when the memory has two defective portions, and this can be readily derived from the concept described in the paragraph [0024]. Accordingly, applicants believe that these limitations are supported by the above-identified paragraph.

3. Rejections of claims under 35 U.S.C. 102:

Claims 13-22 are rejected under 35 U.S.C. 102(e) as being anticipated by Kim et al. (Kim) US Patent No. 6,781,898 or rejected under 35 U.S.C. 102(a) as being anticipated by Chin US Patent Pub. No. 2003/0145250.

Response:

Claim 13 recites that the method for generating a linked list forms a linked list for a memory in which each entry of the linked list corresponds to a portion of the memory, performs a BIST on the memory to identify a first defective portion of the memory, and then updates the linked list to remove from the linked list the entry of the linked list corresponding to the identified first defective portion of the memory.

In the present invention, a temporary linked list is formed in Step 100 (paragraph [0026], lines 2-4), and then a BIST is performed on the header table 20 for storing the linked list in Step 110 to determine which sections of the header table 20 are defective. The operations of Step 120 are described in paragraph [0022], lines 2-6, "each time a defective section is found in the header table 20, the switch 10 will **pause the BIST**, update the linked list dynamically so as not to use the defective section in the list, and **then continue the BIST**." Additionally, the descriptions of Step 140 in paragraph [0024], lines 2-7 state, "each time a defective page is found in

the packet buffer 30, the switch 10 will **pause the BIST**, update the linked list dynamically so as not to use the section corresponding to the defective page of the packet buffer 30, and **then continue the BIST**." According to the above descriptions, it can be appreciated that the "**BIST operation**" and the "**linked list updating operation**" are alternately performed in the exemplary embodiment of the present invention until all sections of the header table and/or all pages of the packet buffer are tested. In other words, the present invention forms a temporary linked list first and then dynamically updates the linked list while performing the BIST operations on the memories.

However, Kim generates the linked list with a different approach. Specifically, Kim performs Defect-Marking-Memory Test process and Defect Row Marking process first, and then performs Defect Row Skipping process to generate the linked list (col. 5 line 65 to col. 6 line 44, and FIG.3). In the Defect-Marking-Memory Test process, the defective rows of the Defect Marking Memory are detected and the addresses of the defective rows are recorded in Defect Address Registers (see col. 6 lines 2-10). In the Defect Row Marking process, the BIST is performed on all memories other than the Defect Marking Memory. When faults of the memories are detected, a Defect Marking Logic is used to write non-zero **marking codes** to the Defect Marking Memory at the defective addresses (see col. 6 lines 17-32). Moreover, as described in col. 6 lines 32-34, "once all addresses are tested, 234, the process shifts to a Defect Row Skipping process." This clearly illustrates that the Defect Row Skipping process for generating the linked list is performed **after** all addresses are tested by BIST according to Kim's disclosure.

Defect Address Registers and Defect Marking Memory are required by Kim to record the detected faulty addresses or marking codes, i.e., the results of BIST. However, since the **claim 13** of the present invention dynamically updates the linked list according to the result of the BIST, Defect Address Registers and Defect Marking Memory are no longer required.

Similar to Kim, Chin performs BIST on all data rows of a packet buffer, **records the test results of all the data rows in a free link table**, and then generates the linked list according to the test results stored in the free link table (blocks 11-13). Specifically, similar to Kim's marking codes, Chin records the blocks of the free link table as "**fail**" or "**good**" according to the BIST results of all the data rows of the packet buffer, and then **reads all the blocks** of the free link table (i.e., reads all the fail/good marks in the free ink table) **when all the blocks are completed with recording** so as to initialize the blocks recorded as "good" in the free link table. That is, the BIST results of **all the data rows** of the packet buffer must be recorded in the free link table **first**, so that the linked list can be generated. In other words, the linked list generation is performed **after** the BIST results of **all the data rows** of the packet buffer are recorded completely. Obviously, Chin's approach is different from the **claim 13** of the present invention. Therefore, the **claim 13** is patentably distinct from both Kim and Chin, and should be allowed.

Claims 14-15 are dependent upon claim 13, and should be allowable if claim 13 is found allowable.

Claim 16 further recites the limitation "the step (c) of updating is

performed **before** the BIST performed in step (b) is completely through with the entirety of the memory". That is, claim 16 limits the operation of updating the linked list is performed **before** the entirety of the memory is completely tested by the BIST. As explained previously, Kim's method generates the linked list **after** all addresses of the memory are tested by BIST, and Chin's method also generates the linked list **after** the BIST results of **all the data rows** of the packet buffer are recorded completely.

It is obvious that the limitation recited in claim 16 of the present invention is patentably distinct from both Kim and Chin due to Kim and Chin failing to disclose the method for updating the linked list **before** the entirety of the memory is completely tested by the BIST. Additionally, claim 16 is dependent upon claim 13, and should be allowable if claim 13 is found allowable.

Claim 17 further recites the limitations "after performing step (c) of updating, continuing the BIST in step (b) to identify a second defective portion of the memory" and "updating the linked list to remove from the linked list the entry of the linked list corresponding to the identified second defective portion of the memory". As explained above, the combination of the limitations in claim 17 and the limitations in claim 13 clearly highlight the "dynamic" nature of the operation of updating the linked list. However, both Kim and Chin generate the linked list **after** the entirety of the memory is completely tested by the BIST but fail to disclose a dynamic updating scheme for the linked list. Additionally, Accordingly, claim 17 is dependent upon claim 13, and should be allowable if claim 13 is found allowable.

Claim 18 is dependent upon claim 13, and should be allowable if claim 13 is found allowable.

Claim 19 recites that the method for generating a linked list forms a linked list for the memory in which the linked list comprises a plurality of entries each having a first pointer field and a second pointer field, the first pointer field for storing a pointer to a corresponding portion of the memory and the second pointer field for storing a pointer to another entry of the linked list; performs BIST on the memory to identify at least one defective portion of the memory; and updates the linked list to remove from the linked list the entry of the linked list corresponding to the identified defective portion of the memory, so that none of the entries of the updated linked list comprises a pointer in the second pointer field that points to the entry corresponding to the identified defective portion.

In contrast to the claim 13, the claim 19 further limits both the contents of the linked list and the updating operations of the linked list. Accordingly, the claim 19 is also patentably distinct from both Kim and Chin, and should be allowed.

Claims 20-21 are dependent upon claim 19, and should be allowable if claim 19 is found allowable.

Claim 22 recites similar limitations as those in claims 17 and 19 to clearly show the "dynamic" nature of the operation of updating the linked list. Since both Kim and Chin fail to disclose or suggest a dynamic updating scheme for the linked list, applicants believe that claim 22 has been placed in condition for allowance.

Applicants respectfully request that a timely Notice of Allowance be issued in this case.

Sincerely yours,

Date: September 18, 2006

Winston Hsu, Patent Agent No. 41,526
P.O. BOX 506, Merrifield, VA 22116, U.S.A.
Voice Mail: 302-729-1562
Facsimile: 806-498-6673
e-mail : winstonhsu@naipo.com

Note: Please leave a message in my voice mail if you need to talk to me. (The time in D.C. is 12 hours behind the Taiwan time, i.e. 9 AM in D.C. = 9 PM in Taiwan.)